STANDARDIZATION DOCUMENT IMPROVEMENT PROPOSAL | 95-15B

INSTRUCTIONS

1. The preparing activity must complete blocks 1,2, 3, and 8. In block 1, both the document number and revision letter should be given.
2. The submitter of this form must complete blocks 4, 5, 6, and 7.
3. The preparing activity must provide a reply within 30 days from receipt of the form.

NOTE: This form may not be used to request copies of documents, nor to request waivers, or clarification of requirements on current contracts. Comments submitted on this form do not constitute or imply authorization to waive any portion of the referenced document(s) or to amend contractual requirements.

I RECOMMEND A CHANGE:	1. DOCUMENT NUMBER **MIL-STD-188-198A**	2. DOCUMENT DATE (YYMMDD) **931215**

3. DOCUMENT TITLE

Joint Photographic Experts Group (JPEG) Image Compression for the National Imagery Transmision Fomat Standard

4. NATURE OF CHANGE *(Identify paragraph number and include proposed rewrite, if possible. Attach extra sheets as needed.)*

This change provides a mechanism to embed Forward Error Corection (FEC) codes inside of the NITF JPEG format. See attached sheets.

Note: This is the OLD APP9 TAG !!

5. REASON FOR RECOMMENDATION

This change request supersedes 95-015A. This change provides protection to the crucial portions of the frame and scan levels within the JPEG format.

6. SUBMITTER

a. NAME *(Last, First, Middle Initial)*	b. ORGANIZATION	
Szwarc, Val	**Lockheed Martin Corporation**	
c. ADDRESS *(Include Zip Code)*	d. TELEPHONE *(Include Area Code)* (1) Commercial **(610) 531-3390** (2) AUTOVON *(If applicable)*	7. DATE SUBMITTED *(YYMMDD)* **970221**

8. PREPARING ACTIVITY **NATIONAL IMAGERY AND MAPPING AGENCY**

a. NAME	b. TELEPHONE *(Include Area Code)* (1) Commercial (2) AUTOVON
c. ADDRESS *(Include Zip Code)*	**IF YOU DO NOT RECEIVE A REPLY WITHIN 45 DAYS, CONTACT:** Defense Quality and Standardization Office 5203 Leesburg Pike, Suite 1403, Falls Church, VA 22041-3466 Telephone (703) 756-2340 AUTOVON 289-2340

5.2.3.3.5.5.4 NITF APP6/(Extension NITF0002) Forward Error Correction (FEC) code. The NITF APP6/(Extension NITF0002) application data segment with an ID string "NITF0002.A" contains the FEC (Forward Error Correction) codes which are used to protect the NITF/JPEG header and misc. table data from bit errors. The ID string follows the form NITFxxxx.V, where xxxx is the extension number and V is the version number. The extension number is 0002 and the current version identifier is A.

The FEC codes are applied to:

a. NITF/JPEG Frame Header and Misc. Tables
b. NITF/JPEG Scan Header and Misc. Tables

Two different forms of the APP6/(Extension NITF0002) application data segment shall be used for each image block, one for the frame overhead data and one for the scan overhead data. The two forms are conditional based on the value of the APP6/(Extension NITF0002) Mode_Type field which discriminates the frame and scan forms of the extension. The frame mode version of the tag is placed in the tag region before the Start of Frame marker code, and the scan version of the tag falls in the tag area preceding each Start of Scan marker in an image block.

The two forms are very similar with the exception of the values contained in the Mode_Type field and the 8 byte ASCII formatted SYNC Code inserted before the Reed-Solomon FEC bytes.

The placement of the APP6/(Extension NITF0002) application segments is shown in figures XX and XXI.

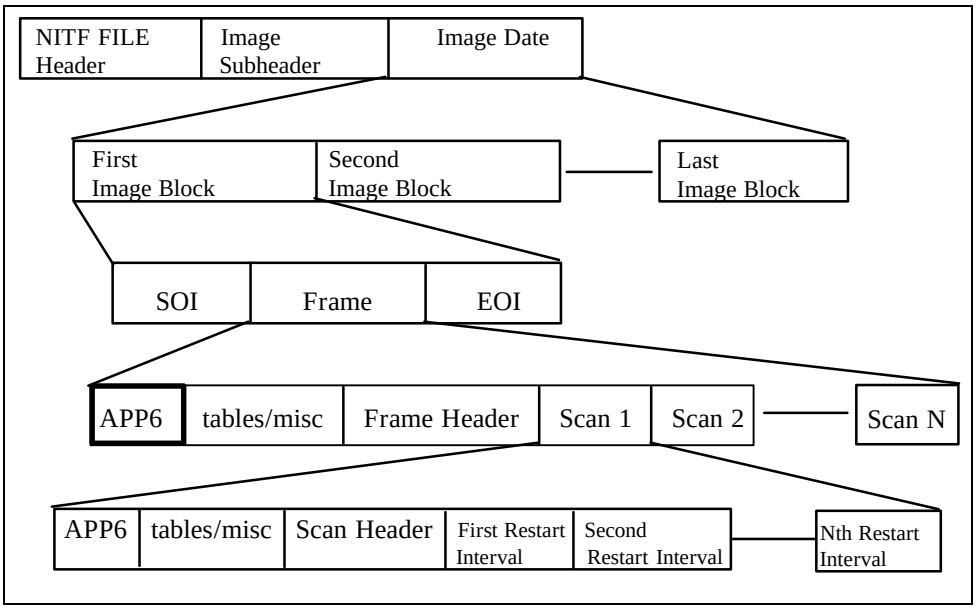

FIGURE XX. NITF 12 bit JPEG/DCT multiple block file structure with FEC (TRANSMISSION MODE = B or P).

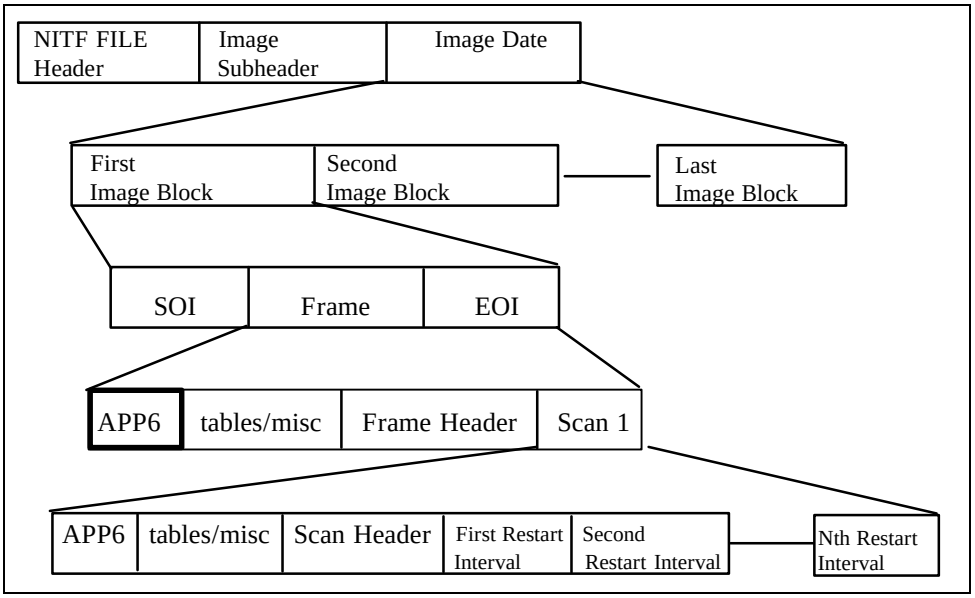

FIGURE XXI. NITF 12 Bit JPEG/DCT multiple block file structure with FEC (TRANSMISSION MODE = S)

The FEC code utilized to error protect the JPEG header and misc. tables is based upon the MIL-STD-2045-44500 (TACO2), paragraph 5.4.2.1, and is termed the FEC-1 code. For purposes of applying the FEC-1 code, the JPEG header and tables shall be logically separated into virtual datagrams of 152 bytes or less in length. For example, to build the datagrams for the Frame level APP6/(Extension NITF0002), the bytes from the SOI marker to the last byte of the Start of Frame (SOF) marker segment shall be used to fill the datagram stream. This includes any other allowed JPEG marker segments or APPs, other than APP6/(Extension NITF0002), present in the tables/misc. area before the SOF marker. The end of the SOF marker segment is the end of the Frame portion of the overhead for the image block. If the number of bytes in the virtual datagram stream is less than or equal to 152 bytes, there will be only 1 datagram. Otherwise, the datagram stream will be divided into 152 byte sections until the last datagram is less than or equal to 152 bytes. Each datagram section of 1 to 152 bytes will produce a 10 byte FEC code from the Reed-Solomon algorithm.

To build the datagram stream for the Scan level APP6/(Extension NITF0002), the bytes following the last byte of the SOF marker segment to the last byte of the Start of Scan (SOS) marker segment shall be included. This includes any other allowed JPEG marker segments or APPs, other than APP6/(Extension NITF0002), present in the tables/misc. area before the SOS marker. The end of the SOS marker segment is the end of the Scan portion of the overhead for the image block. The datagram stream for the Scan overhead is sectioned and processed identically to the Frame overhead stream.

Unlike the TACO2 FEC-1 protocol, the 10 byte FEC codes from all of the datagrams are concatenated within the APP6/(Extension NITF0002) tag, and are not interleaved within the data stream as stated in paragraph 5.4.2.1 of the FEC-1 code in MIL-STD-2045-44500 (TACO2).

The resulting length of the APP6/(Extension NITF0002) application segment will vary dependent upon the number of 10 byte FEC codes in the APP6/(Extension NITF0002) field, which is dependent on the length of the JPEG marker segments and misc. tables present in the overhead.

If the ECC type is set to zero, only the block sizes, SOI offsets, and ASCII SYNC codes are included in the APP6/(Extension NITF0002) extension, and Lp will have a fixed length of 35 bytes.

Table XVIII contains the formats for the APP6/(Extension NITF0002) segment in the Frame and Scan modes.

TABLE XVIII. NITF JPEG APP6/(Extension NITF0002) segment format for NITF JPEG frame and scan level overhead tag.

Offset	Field Value	Field Name	Length	Comments
0	0xFFE6	APP6	2	NITF APP Data Marker
2	Variable	Lp	2	Segment Length (See Note 1)
4	0x4E495446 0x30303032 0x2E41 0x00	ID_STRING	11	Null terminated ID string used to identify the APP6 tag as the ECC overhead protection extension "NITF0002.A"
15	0x02 or 0x03	Mode_Type	1	APP6/(Extension NITF0002) Mode Type 0x02 specifies Frame Mode Code 0x03 specifies Scan Mode Code
16	0x00 or 0x01	ECC Type	1	ECC Type Code specifies the ECC Protection Algorithm ---------- Currently Defined Values: ---------- 0 = No ECC Codes, only offsets and SYNC Code included 1 = RS ECC Codes, SYNC Code, and offsets included
17	(See note 2)	Offset From SOI	4	Offset in bytes from this APP6/(Extension NITF0002) Marker to the SOI Marker of this block
21	(See note 2) or 0x00000000	Previous Block Size	4	Number of bytes in previous image block (See notes 3 & 7)
25	(See note 2)	Current Block Size	4	Number of bytes in this image block (See note 3)
29	0x46454343 0x5353594E or 0x53454343 0x5353594E	Start ECC SYNC Code	8	Frame Sync Code for Start of ECC codes when Mode_Type = 0x02 (ASCII: FECCSSYN) Scan Sync Code for Start of ECC codes when Mode_Type = 0x03 (ASCII: SECCSSYN)
37	Generated (See note 4)	First 10 byte Code	10	10 Byte ECC Code from First Datagram Block
…	…	…	…	…
C	Generated (See note 6)	Last 10 byte Code (See note 5)	10	10 Byte ECC Code from Last Datagram Block

Notes:

1) Length Lp = 35 + (10 * (Number_of_datagrams)) If ECC Type = 0, the Number_of_datagrams = 0.
2) Value is unsigned binary integer format of appropriate length
3) Size in bytes for the image blocks can be calculated by the adding 2 to the offset between the EOI and the SOI markers for each image block. Sizes are stored in binary integer format.
4) Each datagram of 152 bytes or less produces a 10 byte ECC code using the RS FEC protection algorithm.
5) The last datagram of 152 bytes or less produces a 10 byte ECC code using the RS FEC protection algorithm.
6) The offset label of C is used for the conditional offsets dependent on the number of ECC code bytes.
7) The code of 0x00000000 uniquely represents the case when the information regarding the previous block is not included in the tag due to one of the following conditions: a) this is the first block, or b) the image blocks in this NITF product were not sequentially processed and the size was unavailable. This information can be possibly be found in the image block offset tag if it is present in the image subheader.